



Battery parameterisation based on differential evolution via a boundary evolution strategy

Guangya Yang

Center for Electric Power and Energy, Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby 2800, Denmark



HIGHLIGHTS

- Differential evolution with fitness sharing is used for battery model parameterisation.
- A boundary evolution strategy is proposed to update the boundaries of parameters for battery model parameterisation.
- The algorithm addresses the uncertainties of battery modelling from a practical perspective.

ARTICLE INFO

Article history:

Received 24 May 2013

Accepted 17 June 2013

Available online 4 July 2013

Keywords:

Parameter identification
Boundary evolution
Pseudo gradient
Differential evolution

ABSTRACT

Attention has been given to the battery modelling in the electric engineering field following the current development of renewable energy and electrification of transportation. The establishment of the equivalent circuit model of the battery requires data preparation and parameterisation. Besides, as the equivalent circuit model is an abstract map of the battery electric characteristics, the determination of the possible ranges of parameters can be a challenging task. In this paper, an efficient yet easy to implement method is proposed to parameterise the equivalent circuit model of batteries utilising the advances of evolutionary algorithms (EAs). Differential evolution (DE) is selected and modified to parameterise an equivalent circuit model of lithium-ion batteries. A boundary evolution strategy (BES) is developed and incorporated into the DE to update the parameter boundaries during the parameterisation. The method can parameterise the model without extensive data preparation. In addition, the approach can also estimate the initial SOC and the available capacity. The efficiency of the approach is verified through two battery packs, one is an 8-cell battery module and one from an electrical vehicle.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Recent development in the renewable energy field and the electrification of transportation brings attention of electrical engineers to the energy storage technologies. As an important energy carrier, batteries are seen as one of the key components in the future energy system. In terms of the electro-chemical batteries, currently the main types include lithium-ion, Lead-acid and Nickel–cadmium batteries. Among these batteries, Lithium-ion batteries are seen as one of the most promising technologies for the high power and energy density. Accurate modelling of Lithium-ion batteries can confirm the applications and control strategies as it will provide useful information such as the state of charge (SOC), state of health (SOH) and electric characteristics. Therefore, the model of Lithium-ion batteries has been investigated by researchers in the last several decades.

The Lithium-ion battery models may consist of two classes, electrochemical models [1–4], and equivalent circuit models [5–8]. The electrochemical model describes the complex chemical process, and is mainly expressed by partial differential equations (PDEs). This type of model can provide remarkable accuracy under certain conditions [4,9]. However, the model verification can be difficult as the solution of PDEs is highly dependent on the initial and boundary conditions.

The equivalent circuit models use electric circuits, typically resistance–capacitance (RC) circuits, to represent the voltage–current (V – I) characteristics of the battery with high accuracy [10,6]. Dependent on the complexity, it can be further divided into internal resistance model, Thevenin and improved Thevenin models [11]. Comparing to the electrochemical models, equivalent circuit models are simpler as the RC circuit can be expressed by ordinary differential equations (ODEs), which makes it easy to be verified and implemented in practice. The accuracy of the model can reach up to 95% [6], and can be improved with the increasing

E-mail address: gyy@elektro.dtu.dk.

order of the circuits. As to the simple formulation and the high accuracy, this type of model potentially can be adopted for real world use.

Besides the cell model, the battery pack model is also investigated in the literature. The studies in Refs. [8,12,13] show that it is possible to obtain a battery pack model via scaling-up the single cell models. Other than the aspect of V - I characteristics, battery models based on other aspects are also developed, e.g., thermal characteristics [14], life time [15], capacity fade [16], etc. These models are usually established for specific applications, however they can also be combined with the V - I models to enhance the model accuracy [17].

Battery parameterisation is a necessary procedure in battery modelling. This procedure requires data preparation and model validation. At the data preparation stage, usually a set of laboratory experiments will be performed, such as charging/discharging test, impedance spectroscopy test, open circuit voltage (OCV) test [18] and calendar/cyclic life test [19], to name a few. Based on the test results, methods can be applied to tune the model parameters to obtain the best match between the model outputs and the measurements. It is a typical nonlinear curve fitting problem. So far, identification methods applied include least square (LS) method [20], Kalman filtering [21], subspace method [7], etc. Those conventional methods usually assume that the model parameters are invariant. However, certain battery model parameters can be variant with respect to the level of SOC and SOH, therefore those traditional methods may encounter problems to tune accurately the battery parameters. Instead, methods from artificial intelligence (AI), such as genetic algorithms (GAs), start to attract the attention from battery modelers [22]. This type of method is problem-independent and has high flexibility in accommodating different optimisation formulations. However, modifications are still required to cater to different applications.

The battery models are expected to provide the information such as the battery SOC, available capacity, and internal impedance, etc. These information can be further used as the inputs for different operation functions in the battery management system (BMS). However, the battery behaviour is affected by many factors, such as the life time and the ambient environment, therefore the models of batteries may change time to time. The following situations may be encountered in practice,

- The battery OCV, SOC, as well as the internal impedances will change with temperature, and degrade along the time of use, as to the loss of calendar life and cyclic life;
- A battery pack may be upgraded partly or fully, in the case of having damaged cells, capacity expansion, swapping with another pack, or change of electrical configuration, etc.

Such cases may result from the applications of electrical vehicles or stationary storages for cooperation with renewable generation

units. Following the changing characteristics and the electric configurations of the battery packs, it could be necessary to periodically verify and update the model [4,21,23,24]. In this paper, a parameterisation scheme is developed based on differential evolution (DE) for the equivalent circuit model of Lithium-ion battery. DE is from the family of evolutionary algorithms (EAs), and featured as an optimiser of fast, robust, and easy to implement. In this work, a DE with fitness sharing is employed to enhance the optimisation efficiency.

One of the difficulties in the parameterisation is to identify the boundaries of the parameters. As far as to the author's knowledge, this problem has not been investigated in the literature. In this work, a boundary evolution strategy (BES) is proposed specifically for the battery parameterisation. The efficacy of the strategy is first verified through a mathematical function. The strategy is then incorporated into the modified DE for battery parameterisation. The proposed approach can be conveniently applied without extensive data preparation. Furthermore, the approach can also be used to estimate the initial SOC and the available capacity. It is worth mentioning that the algorithm can also be easily adopted to verify the models of other types of batteries.

The paper is organised as the following. In Section II, the adopted equivalent circuit model is described. Section III gives a short introduction to the DE with fitness sharing. The BES is described in Section IV. Detailed case studies are presented in Section IV, followed by discussion and conclusion.

2. Equivalent circuit model of lithium-ion batteries

A typical layout of the circuit model of the lithium-ion batteries comprises of the SOC part and the impedance part, as shown in Fig. 1. The SOC part provides the battery SOC by using the battery capacity, charging/discharging current, and self-discharge resistance. It may be expressed by,

$$\text{SOC} = \text{SOC}_0 + \frac{1}{Q} \int (I_{\text{charging/discharging}} - I_{\text{self-discharging}}) dt \quad (1)$$

where the SOC_0 , SOC are the initial SOC and the current SOC, respectively. Q is the capacity of the battery in Ampere-Hour(Ahr), $I_{\text{charging/discharging}}$ and $I_{\text{self-discharging}}$ represent the charging/discharging current and self-discharging current, respectively.

The SOC is used as input to the impedance part of the circuit model. This part comprises of a controlled voltage source which represents the battery OCV, and internal impedances consisting of a series resistance and several RC circuits. The impedance part is used to describe the relation between the charging/discharging current and the battery terminal voltage. When battery is subject to a sudden change of charging/discharging current, the series resistance describes the instantaneous terminal voltage drop while the RC circuits provide the short-term and long-term variations of the

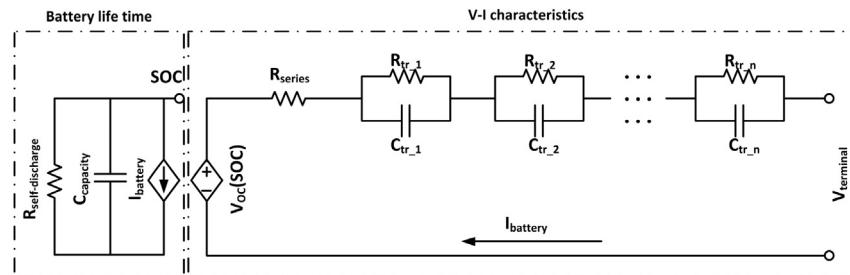


Fig. 1. Typical equivalent circuit model of Lithium-ion batteries.

terminal voltage. The model accuracy increases with the increase of the number of RC circuits, however the computational efficiency will decrease accordingly.

2.1. Battery circuit model formulation

The circuit model can be represented in Laplacian domain. For one RC circuit, the impedance in Laplacian domain can be expressed by

$$Z(s) = \frac{R}{1+RCs} \quad (2)$$

Therefore for a circuit with n RC circuits as shown in Fig. 1, the total Laplace impedance can be expressed by,

$$Z(s) = \frac{R_{tr_1}}{1+R_{tr_1}C_{tr_1}s} + \frac{R_{tr_2}}{1+R_{tr_2}C_{tr_2}s} + \dots + \frac{R_{tr_n}}{1+R_{tr_n}C_{tr_n}s} + R_{\text{series}} \quad (3)$$

For the sake of computational efficiency, the circuit used in this work employs two RC circuits with one series resistance. The equivalent impedance Z_{eq} can then be expressed by,

$$Z_{\text{eq}}(s) = \frac{a_1 s + a_0}{b_2 s^2 + b_1 s + b_0} + c \quad (4)$$

where

$$\begin{cases} a_1 = R_{tr_1}R_{tr_2}C_{tr_1} + R_{tr_1}R_{tr_2}C_{tr_2} \\ a_0 = R_{tr_1} + R_{tr_2} \\ b_2 = R_{tr_1}R_{tr_2}C_{tr_1}C_{tr_2} \\ b_1 = R_{tr_1}C_{tr_1} + R_{tr_2}C_{tr_2} \\ b_0 = 1, \\ c = R_{\text{series}} \end{cases} \quad (5)$$

The voltage drop along the battery internal impedance therefore can be expressed by,

$$V_{\text{drop}}(s) = Z_{\text{eq}}(s)*I_{\text{charging/discharging}}(s) \quad (6)$$

where the $V_{\text{drop}}(s)$, $I_{\text{charging/discharging}}(s)$ represent the internal voltage drop and the charging/discharging currents of the battery in Laplace domain, respectively. By taking $I(s)$ as input and $V(s)$ as output, the above system can be transformed back to time domain in controllable canonical form,

$$\dot{x}(t) = \begin{bmatrix} -b_1/b_2 & -1/b_2 \\ 1 & 0 \end{bmatrix}x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix}I_{\text{charging/discharging}}(t) \quad (7)$$

$$V_{\text{drop}}(t) = [a_1 \ a_0]x(t) + [c]I_{\text{charging/discharging}}(t)$$

where $x(t)$ represents the state variables of the system. $I_{\text{charging/discharging}}(t)$, $V_{\text{drop}}(t)$ are the charging/discharging current, and the voltage drop on the internal impedance in time domain, respectively. The battery terminal voltage $V_{\text{terminal}}(t)$ can then be expressed by using the open circuit voltage $V_{\text{OCV}}(t)$ subtracts the voltage drop $V_{\text{drop}}(t)$ on the battery internal impedance,

$$V_{\text{terminal}}(t) = V_{\text{OCV}}(t) - V_{\text{drop}}(t) \quad (8)$$

Substituting the above equation into Eq. (7), a full battery model can be obtained,

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} -b_1/b_2 & -1/b_2 \\ 1 & 0 \end{bmatrix}x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix}I_{\text{charging/discharging}}(t) \\ V_{\text{terminal}}(t) &= -[a_1 \ a_0]x(t) - [c]I_{\text{charging/discharging}}(t) + V_{\text{OCV}}(t) \end{aligned} \quad (9)$$

2.2. Formulations of model parameters

The values of resistance, capacitance and the battery OCV can be affected by a number of factors, SOC, SOH, ambient temperature, charging/discharging currents, etc. In this paper, focus is given to the impact from SOC on the OCV and circuit elements. The impacts from SOH and ambient temperature may be mitigated by periodically updating the battery model parameters. The OCV is formulated as a polynomial function of SOC,

$$V_{\text{OCV}} = \sum_{i=1}^n P_i^{\text{OCV}} \text{SOC}^i \quad (10)$$

where P_i^{OCV} is the coefficient of the i -th term in the OCV function. The battery impedances are formulated as below,

$$\begin{aligned} R_{\text{series}} &= P_1^{\text{rs}} e^{P_2^{\text{rs}} \text{SOC}} + P_3^{\text{rs}}; \\ R_{tr_1} &= P_1^{\text{r1}} e^{P_2^{\text{r1}} \text{SOC}} + P_3^{\text{r1}}; \\ C_{tr_1} &= P_1^{\text{c1}} e^{P_2^{\text{c1}} \text{SOC}} + P_3^{\text{c1}}; \\ R_{tr_2} &= P_1^{\text{r2}} e^{P_2^{\text{r2}} \text{SOC}} + P_3^{\text{r2}}; \\ C_{tr_2} &= P_1^{\text{c2}} e^{P_2^{\text{c2}} \text{SOC}} + P_3^{\text{c2}}; \end{aligned} \quad (11)$$

where the $P_x^{\text{rs}}, P_x^{\text{r1}}, P_x^{\text{c1}}, P_x^{\text{r2}}, P_x^{\text{c2}}$, $x \in \{1, 2, 3\}$ are the coefficients in the expressions of the circuit elements. The battery parameterisation is basically to find out the values of these coefficients. The OCV parameters can be handily estimated by the OCV test results beforehand.

In addition, for the SOC part of the model, it is usually assumed that the initial SOC and the battery available capacity are known beforehand. It can true for the laboratory tests, as the battery are well monitored with steady ambient environment. However, it could be challenging to obtain the exact values in real life due to variant ambient conditions or monitoring errors. It is of practical importance to have an estimation regarding the initial battery condition. The approach proposed in this paper can be extended to include the initial SOC and battery capacity, thanks to the flexibility of EAs.

3. DE with fitness sharing

As the parameters in Eq. (9) are varying with SOC, the model represents a linear parameter varying system where no analytical solution can be obtained. In this work, the advances of differential evolution from the family of EAs are exploited. DE is first proposed by Storn and Price at Berkeley over 1994–1996 as a novel approach to the numerical optimization problems [25,26]. Besides the common advantages from EAs, such as handling non-continuous, non-convex and non-differential problems and giving global or near global optimal solutions, DE is featured by fast, robust, and easy to implement, and has been vastly applied among different engineering fields [26,27].

3.1. Brief introduction to canonical DE

Similar as other EAs, DE is also an algorithm of population-based, iteration-driven and steered search. For the G -th iteration, a DE population contains number NP of n -dimensional vectors X_i , $i \in \{1, 2, 3, \dots, NP\}$, where each vector $X_i \in R^n$ represents a candidate solution of the optimization problem. Using x instead of X_i , the optimization problem can be expressed by,

$$\min f(x), x \in \Omega \quad (12)$$

where $\Omega \subset R^n$ is the solution set defined by the constraints, and $f(x): \Omega \rightarrow R$ is a real-valued function to be optimised, which is used for fitness calculation. In one iteration, DE performs the operations mutation, recombination, and selection to produce the population of next iteration to drive the optimization process. These operators are briefly introduced here.

3.1.1. Mutation and recombination

The mutation operator exploits the differences among the vectors to produce mutated vectors, and a nonuniform recombination operator is attached with this operator to generate trial vectors. They are implemented and finished almost simultaneously. A trial population containing NP trial vectors will be obtained afterwards. The i -th mutated vector X_i^m is obtained by the following equation:

$$X_i^m = X_{r1} + c^*(X_{r2} - X_{r3}) \quad (13)$$

where $r1 \neq r2 \neq r3$ are randomly selected locations in the population, and the values are different with the running index i . c is a positive constant scalar. This operator is combined with a recombination operator, which exchanges the elements between the vector X_i in the current population and the mutated vector X_i^m to produce a trial vector. A probability index CR is preset to control the recombination. Firstly, a random number $rand$ is generated within $[0, 1]$, and if $rand \neq CR$, then the j -th element x_{ij} of the trial vector X_i^T will employ the j -th element from the mutated vector X_i^m ; otherwise use the element from X_i . To enforce this operator, the probability CR at certain entries may be defined at 1. A pseudo code of the mutation and recombination process is given below.

To generate the i -th trial vector

Step 1: Generate three random indices $r1, r2, r3$, where $r1 \neq r2 \neq r3 \neq i$;

Step 2: Generate an n -dimensional vector V^{rand} , where all the entries are randomly generated within $[0, 1]$;

Step 3: If the k -th element of V^{rand} , $V_k^{rand} \leq CR$, $k = 1, 2, 3, \dots, n$, then the k -th element of the i -th trial vector X_i^T will be:

$$X_i^T(k) = X_{r1}(k) + c^*(X_{r2}(k) - X_{r3}(k)) \quad (14)$$

Otherwise the value of the k -th entry of X_i will be chosen,

$$X_i^T(k) = X_i(k) \quad (15)$$

3.1.2. Selection

In the selection operator, the trial vector X_i^T will be presented to the fitness function, and if the fitness value is better than the original vector X_i , then X_i^T will be selected to the next population and appear in the next iteration, otherwise X_i will survive.

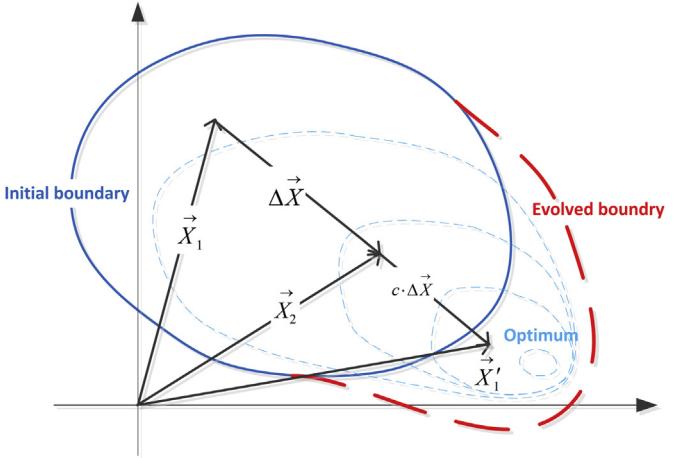


Fig. 2. An illustration of problem where a better solution is out of the predefined solution space.

3.2. DE with fitness sharing

One of the issues of DE is to maintain the population diversity, especially when the population is approaching the vicinity of the optimum [28]. One of the efficient techniques proposed in the literature is to introduce a sharing function in the fitness calculation to penalise the vectors closing to other vectors [29,30]. In order to define the shared fitness, two functions need to be defined, a non-negative distance function $\gamma(X_i, X_j)$, which measures the similarity between two vectors, and a sharing function $sh(\gamma)$, which provides a penalty factor to the fitness of the vector. The sharing function shall fulfil

$$sh(\gamma) = \begin{cases} 0 \leq sh(\gamma) \leq 1 \\ sh(0) = 1 \\ \lim_{\gamma \rightarrow \infty} sh(\gamma) = 0 \end{cases} \quad (16)$$

A “niche radius” σ is defined which penalises the fitness of similar individuals. The sharing function is defined as,

$$sh(\gamma) = \begin{cases} 1 - (\frac{\gamma}{\sigma})^\alpha, & \text{if } 0 \leq \gamma \leq \sigma \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

where α is an arbitrary factor dependent on the problem. In DE, the evolution process aims at searching for the minimum optimal values. Therefore, the sharing fitness is calculated via using original

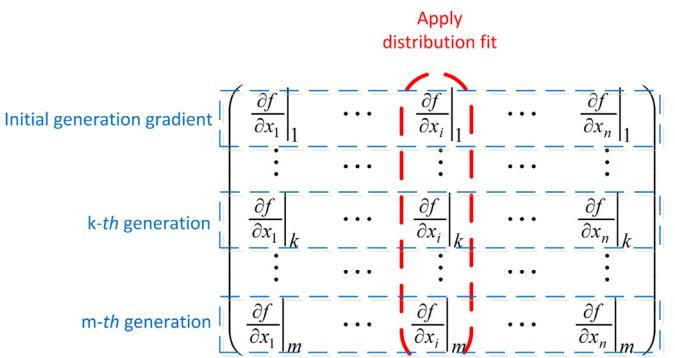


Fig. 3. A demonstration of the gradient estimation process.

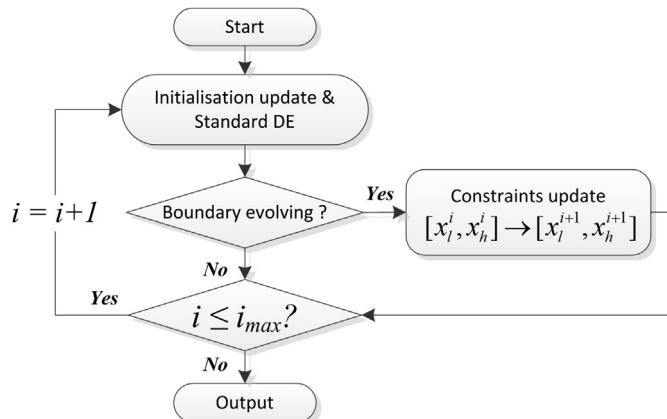


Fig. 4. The flowchart of the modified DE algorithm.

fitness times the sharing factor so that the fitness values of the similar vectors are increased.

$$f_{sh} = f_i \cdot \sum_{j=1}^{NP} sh_j(\gamma(X_i, X_j)) \quad (18)$$

3.2.1. Optimisation formulation

Finally, the objective function of the parameterisation is,

$$f = \min \sum (V_{\text{measurement}} - V_{\text{terminal}})^2 \quad (19)$$

which minimises the mismatch between the voltage measurements $V_{\text{measurement}}$ at the battery terminal and the simulated terminal voltage V_{terminal} using current measurements $I_{\text{charging/discharging}}$ as input.

4. Boundary evolution strategy

One of the uncertainties in optimisation is defining the variable limits. As the equivalent circuit model is a conceptual mapping of the physical model, and employs a large number of parameters with highly variant values, the parameterisation is inherently a

complex problem where the optimum point is likely located outside of the initially defined boundaries. The problem can be illustrated from Fig. 2, where the generated new vector X'_1 is towards the optimum however is forced within the defined boundary. This will affect the optimality of the final solution. Therefore, it is necessary to adaptively adjust the boundaries to improve the search capability, which seems have not been fully investigated in the literature. In this work, a boundary evolution strategy is proposed to evolve the boundary to the optimum, therefore enhance the optimisation capability of the algorithm as well as the optimality of the results.

The problems handled by EAs are usually non-differential, non-continuous and multi-modal, and hence there is no explicit gradient expression of the objective function. Here in this work, a method is proposed to estimate the gradient of the objective function exploiting the differences among the vectors. The concept of pseudo gradient is adopted and further developed [31]. By estimating the gradient, the necessity of boundary adjustment can be determined after one run of DE.

For a multivariable function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, the total difference is defined by,

$$\Delta f(x) = \sum_{i=1}^n f_{xi} \Delta x_i \quad (20)$$

where the f_{xi} refers to the partial derivative on the i -th variable. In implementation, the values of $\Delta f(x)$ and Δx_i can be obtained by using a pair of vectors subtracting each other. In order to obtain the partial derivative f_{xi} , n independent equations are required. To ensure this, $m, m > n$ pairs of vectors should be selected from the population to obtain m equations. In matrix form, we have

$$\Delta f(x) = A \nabla f(x) \quad (21)$$

As the dimension of matrix A is $m \times n, m \neq n$, the pseudo gradient $\nabla f(x)$ can be further calculated by,

$$\nabla f(x) = (A^T A)^{-1} A \Delta f(x) \quad (22)$$

This calculation can be performed in every iteration in DE. It is worth mentioning that since DE is a blind search technique, there is

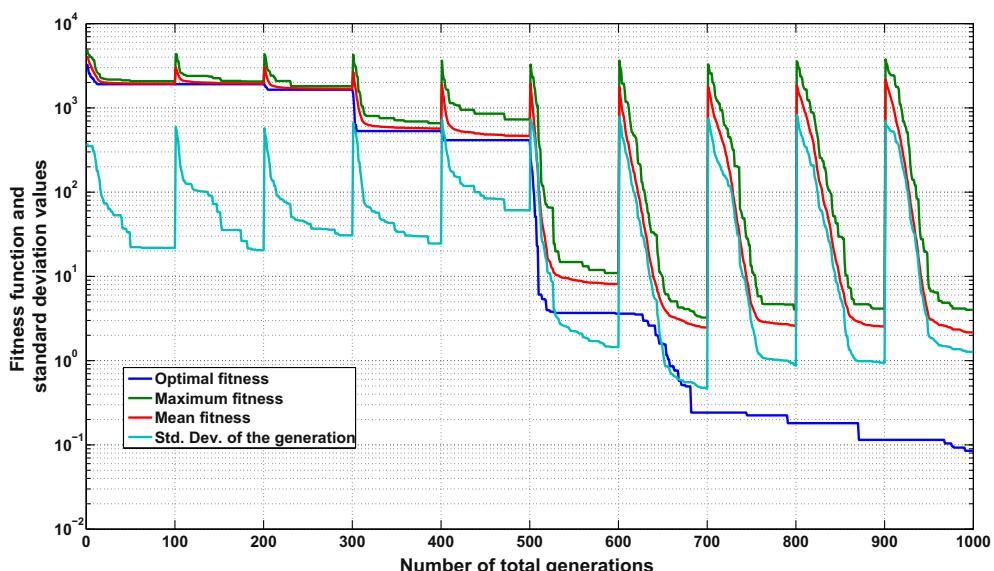


Fig. 5. The evolving process of DE solutions.

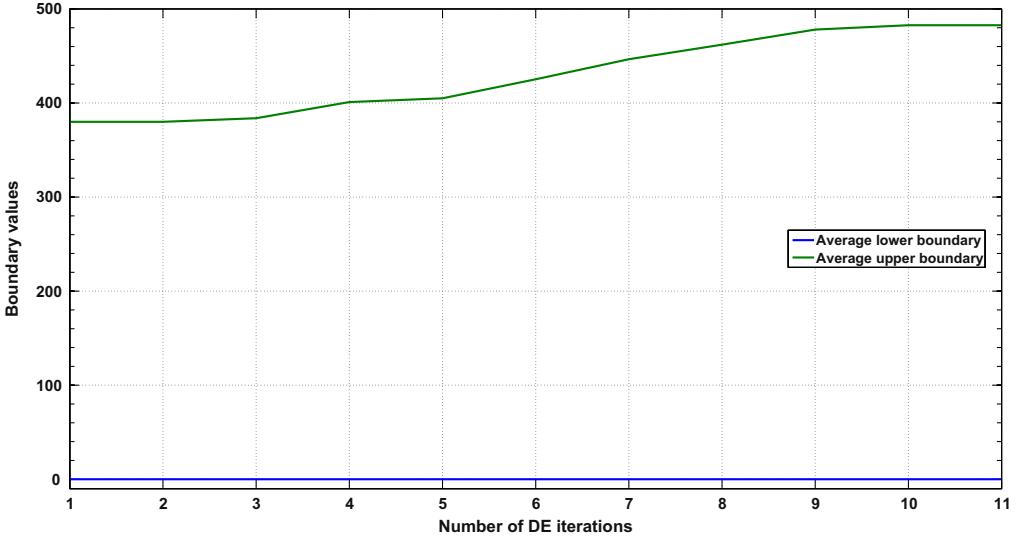


Fig. 6. The evolving process of the variable boundaries.

inherent randomness in the searching process. This will impose the value of the pseudo gradient. For instance, at the initial iteration stage, the mathematical distances among the vectors are high, which may generate large absolute values on $\Delta f(x)$ and $\nabla f(x)$. While at the end of iteration, as the population is close to convergence, the distances among the vectors are low. This may result in large absolute values at some entries in the $(A^T A)^{-1}$ matrix and therefore give rise to large absolute value of $\nabla f(x)$. It is also possible that at some iteration the pseudo gradient value cannot be calculated, since there are no n independent equations can be formulated. Therefore, the pseudo gradient may not be suitable to be directly used for further analysis.

To resolve this issue, probability technique is employed here to further refine the pseudo gradient information. It is assumed that each entry in the pseudo gradient vector follows normal distribution along the iteration process. For the i -th variable x_i , the $\partial f/\partial x_i$ calculated over the iteration process are fit into normal distribution, and the mean value is taken as the actual value of the partial derivative $\partial f/\partial x_i$. The gradient of the objective function can be estimated after one run of DE. This procedure is demonstrated in Fig. 3.

An evolution procedure of the variable boundaries is formulated hereafter. If the i -th parameter value is at the vicinity of the upper limit and the partial derivative $\partial f/\partial x_i$ is positive, then the upper limit of x_i needs to be increased; on the contrary, if the i -th parameter is at the vicinity of the lower limit while the partial derivative is negative, the lower boundary of x_i will need to be decreased. Two conditions need to be fulfilled for boundary evolution,

1. Some entries in the obtained optimum are at the vicinity of their boundaries;
2. Valid pseudo gradient information;

The equations used to update the boundaries are expressed below,

$$\begin{cases} x_{l,i}^{\text{new}} = x_{l,i}^{\text{old}} - \alpha(x_{u,i} - x_{l,i}) \\ x_{u,i}^{\text{new}} = x_{u,i}^{\text{old}} + \alpha(x_{u,i} - x_{l,i}) \end{cases} \quad (23)$$

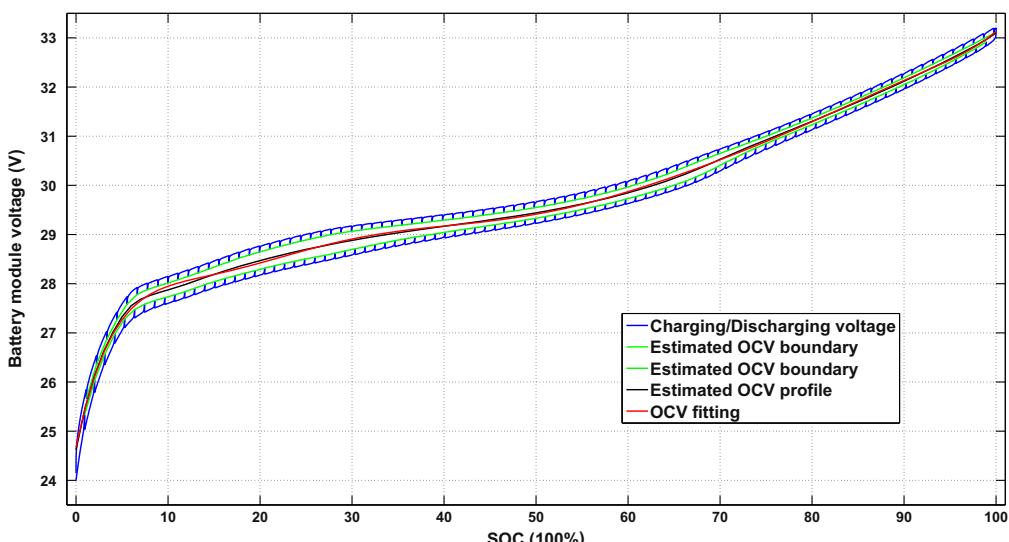


Fig. 7. The OCV test result for NMC battery module.

Table 1

Parameterisation results from NMC battery module.

Parameters	Initial SOC	Capacity	P_1^{rs}	P_2^{rs}	P_3^{rs}
Charging	6.83×10^{-4}	77.71	6.38×10^{-3}	-0.442	3.81×10^{-3}
Discharging	1	77.85	0	0.168	7.15×10^{-4}
Parameters	P_1^{r1}	P_2^{r1}	P_3^{r1}	P_1^{c1}	P_2^{c1}
Charging	1.33×10^{-5}	0.206	3.78×10^{-4}	5.85×10^{-4}	0.274
Discharging	-4.05×10^{-3}	3.65×10^{-6}	5.45×10^{-3}	2.11×10^{-4}	0.141
Parameters	P_1^{r2}	P_2^{r2}	P_3^{r2}	P_1^{c2}	P_2^{c2}
Charging	1.71×10^{-3}	-9.60×10^{-7}	2.06×10^{-2}	2.02×10^{-3}	5.34×10^{-3}
Discharging	9.93×10^{-6}	0.293	1.89×10^{-2}	1.17×10^{-3}	0.345

The $x_{l,i}$ and $X_{u,i}$ are the lower and upper boundaries of variable x_i , respectively. α is a constant scalar which can be dependent on the problem.

4.1. Incorporate BES into DE

It is worth mentioning that the above strategy can be integrated into any EAs. For DE, this procedure can be handily integrated into the selection operator. During the selection operation, two vectors from the current and trial population respectively are compared by the fitness values. At the same time the Eq. (20) can be preceeded. For a population with size NP , equations can be prepared for the calculation of Eq. (22). As normally $NP \gg n$, the characteristic matrix $A^T A$ usually is invertible unless the population is close to convergence. Besides, the minimum eigenvalue of $A^T A$ can be used as an indicator of the maturity of the population.

The whole procedure for the modified DE is shown in Fig. 4. The algorithm is validated through two standard testing functions, Schwefel and Griewank [32], which are commonly used for testing the efficiency of EAs. Both cases show that the BES significantly improves the search capability of DE. For the sake of brevity, only the results of Schwefel function are presented here. The Schwefel function is described as below,

$$f(x_i) = 418.9829n + \sum_{i=1}^n (-x_i \sin(\sqrt{|x_i|})) , \quad n = 10 - 500 \leq x_i \leq 500, i \in \{1, \dots, 10\} \quad (24)$$

The optimal point occurs at $f(x_i) \approx 0$, where $x_i \approx 420.9687$, $i \in \{1, \dots, 10\}$. To test the efficacy of the proposed approach, the initial

variable boundaries are purposely set at $x_i \in [0.380]$, and the population size, total generation of DE are set at 50, 100 respectively. The number of runs of DE is set at 10, and the value of α in Eq. (23) is 0.1. The best value obtained by the last run of DE will be used to initialise the population of the next DE trial. The first condition of boundary evolution is formulated by,

$$\begin{cases} (x_{opt,i} - x_{l,i}) / (x_{u,i} - x_{l,i}) \leq 0.1, \text{ or} \\ (x_{u,i} - x_{opt,i}) / (x_{u,i} - x_{l,i}) \leq 0.1 \end{cases} \quad (25)$$

The $x_{opt,i}$ is the optimal value of variable x_i . If any of the above equations is valid, the boundary of x_i will need to be updated. The approach is tested by several trials and it is found that BES is able to provide robust results. A typical result of the iteration and the boundary evolving procedure is shown in Figs. 5 and 6.

The final result of x is very close to the theoretical optimum. It can be seen from Fig. 5 that the evolution of the boundary clearly improves the quality of the solution. As all the variable boundaries have similar magnitudes, therefore only the mean values of the boundaries are shown in Fig. 6, which depicts the typical evolution of the boundaries. It can be seen that only the upper boundary values are changed while the lower boundaries are kept constantly. This matches the pre-knowledge about the optimum. The evolution of the upper variable limits stops once the optimum is bracketed.

5. Case studies

The above approach is applied to parameterise two battery models at the pack level, a 75 Ah 8-cell Lithium–Nickel–Manganese–Cobalt–Oxide (NMC) battery module, and the battery of a real electric vehicle (EV) with capacity of 100 Ah, 35 kWh. The results are hereby organised into two sections.

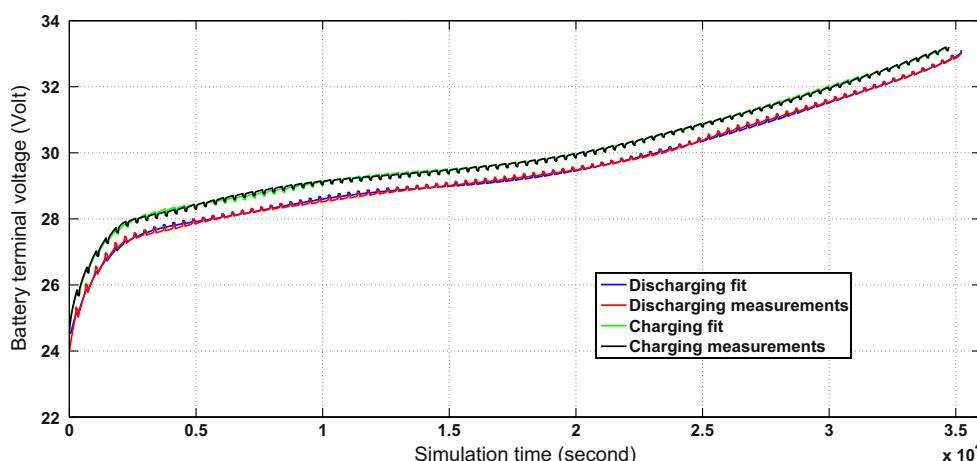


Fig. 8. Comparison of NMC module model outputs with measurements.

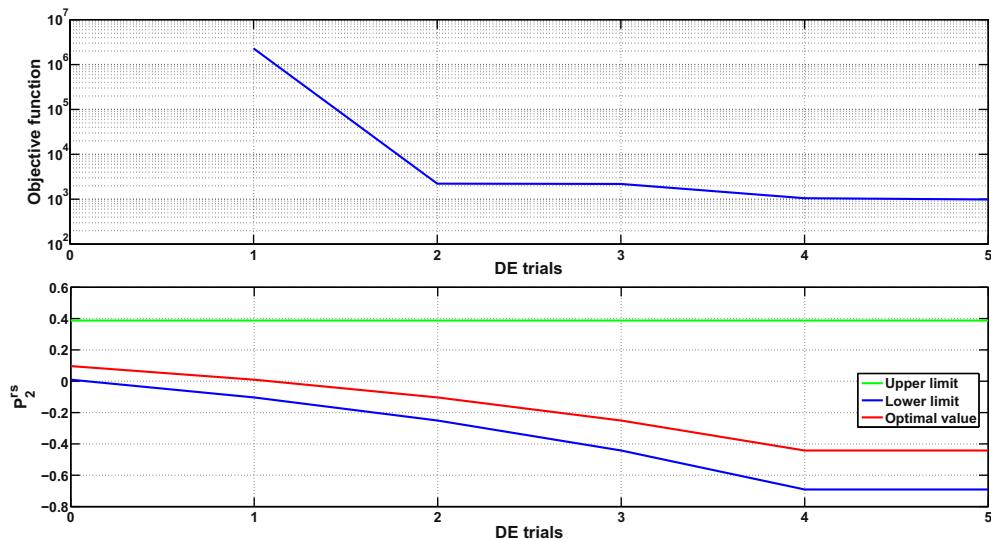


Fig. 9. An example of the evolving process of the optimal parameter value and the boundaries.

5.1. NMC battery pack parameterisation

The parameterisation of NMC battery pack is based on laboratory experiments. The results from OCV test are exploited to parameterise the model. The OCV test uses the method proposed in Refs. [18], where the battery pack is experienced a full charging–discharging cycle. Constant currents are used, and pauses of 1 min are applied during the charging–discharging cycle for every 5 min. During the pause, the battery voltage will either decrease (charging) or increase (discharging) towards the OCV. The OCV curve lies in the middle of the boundaries estimated by using the lowest voltages (charging) and highest voltages (discharging) measured in the pauses. The test is carried out under ambient temperature 22 °C.

The current applied for charging and discharging is constantly 10 A. Before the full charging–discharging cycle, the module is charged to 100% SOC. The battery is first charged with 10 A for 5 min and the current is reduced to 0 A for 1 min. As soon as one of the eight cells in the module reaches the maximum allowable charging voltage of 4.15 V, the current is interrupted and the

current SOC is defined as 100%. Immediately after this, the module is discharged with 10 A for 5 min. Then the battery current is set to 0 A for 1 min. This step is repeated until one of the eight cells in the module reaches the minimum allowable discharging voltage 3 V. The module is charged again in steps of 5 min with 10 A and 1 min with 0 A, until one of the eight cells reaches 4.15 V. The OCV results and the fitted model output are shown in Fig. 7.

The measurements from the OCV test can be further used for parameterisation, since the dynamic VI characteristics can be disclosed during the applied pauses. The parameters of the battery are validated for charging and discharging separately since the battery may have different parameters in each case. The DE population size and the number of iterations are set at 50, 30, respectively, with total 5 DE runs. Detailed results are listed in Table 1. The performance of the generated battery models for both charging and discharging are depicted in Fig. 8.

From Table 1, it is seen that the approach provides similar estimation on the total capacity in both cases. The initial SOC estimation follows the test design. The parameters of the RC circuits are different, which shows the battery has different models during

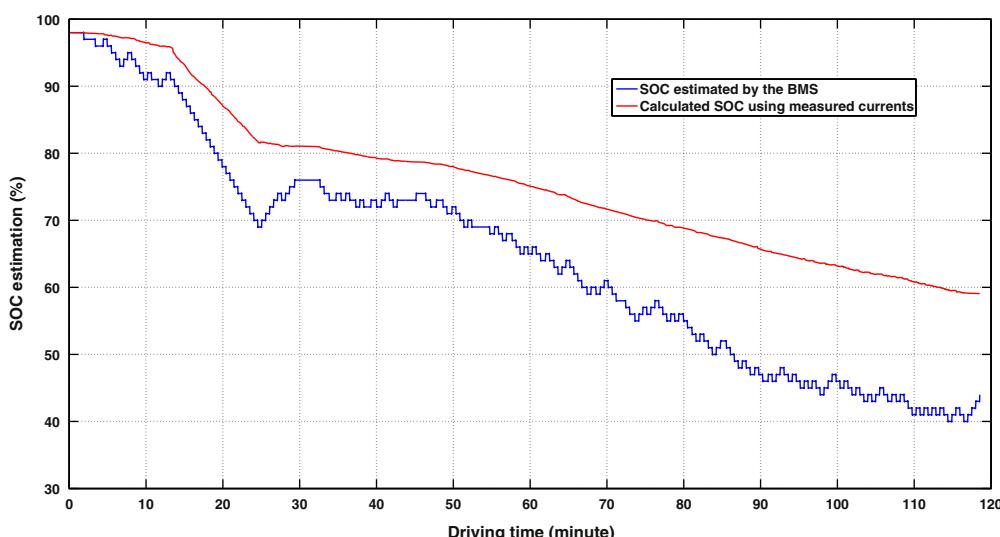


Fig. 10. The mismatch between the BMS estimation and the calculation.

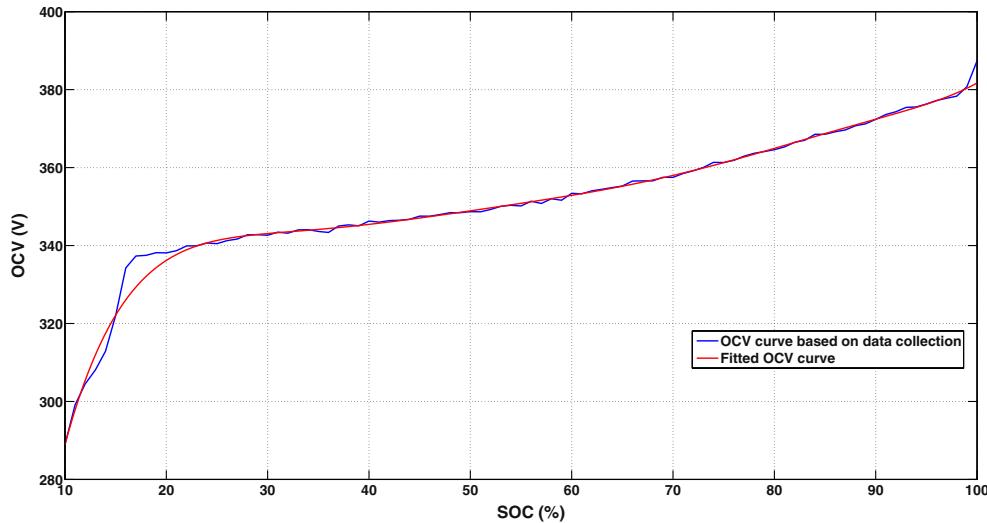


Fig. 11. The fitted OCV curve.

charging and discharging. Fig. 8 compares the mismatch between the model and the measurements. It is clearly seen that the models provide sufficient accuracy in both cases.

The efficacy of the proposed BES is shown in Fig. 9. For the sake of brevity, only the parameter P_{rs}^2 is chosen to illustrate the BES process. The initial boundary of P_{rs}^2 is $[9.7 \times 10^{-3}, 0.39]$, while after 5 DE trials, the lower boundary is decreased to -0.69 . It can be seen that the proposed algorithm efficiently updates the variable boundaries to improve the optimum.

5.2. Electric vehicle battery parameterisation

Another battery is from a real EV with an approximately 100 Ahr (35 kWh) Lithium-ion battery pack. The data procured are recorded by the EV battery management system (BMS) for around 1 year. The data include the battery current and voltage, temperature, estimated SOC, battery capacity, etc. There is no battery model provided by the car manufacturer beforehand. From experience, it is found necessary to establish a battery model to validate the BMS performance, e.g., the SOC and the battery capacity, which is crucial to exploit the storage capability of EVs to facilitate the renewable energy integration. Fig. 10 shows a typical example of

the mismatch between the estimated SOC during EV driving and calculated SOC by using battery current integration. There is approximately 15% difference between the two values after around 2 h driving. A good estimation of SOC, battery capacity, and internal impedance can be useful to improve the efficiency of battery management system.

To establish the battery model, the initial step is to construct the OCV model. This step is achieved by using the recorded voltage measurements and SOC estimated by the BMS at the points where the car is still for some time without charging and discharging activities. It is fair to use those voltages as the OCV. By aligning the voltage measurements and the corresponding SOC, an estimated OCV can be obtained. The OCV curve is also verified by the voltage measurements during constant current charging and discharging using the principle that the OCV locates between charging and discharging terminal voltages. The parameters in Eq. (10) can be obtained by fitting this curve. The results of the OCV estimation are shown in Fig. 11.

The parameterisation chooses two sets of measurements during driving. The DE is set by population size 50 with 50 iterations. 5 DE trials are employed for BES. Comparisons between the model outputs and the measurements are shown in Figs. 12 and 13. An

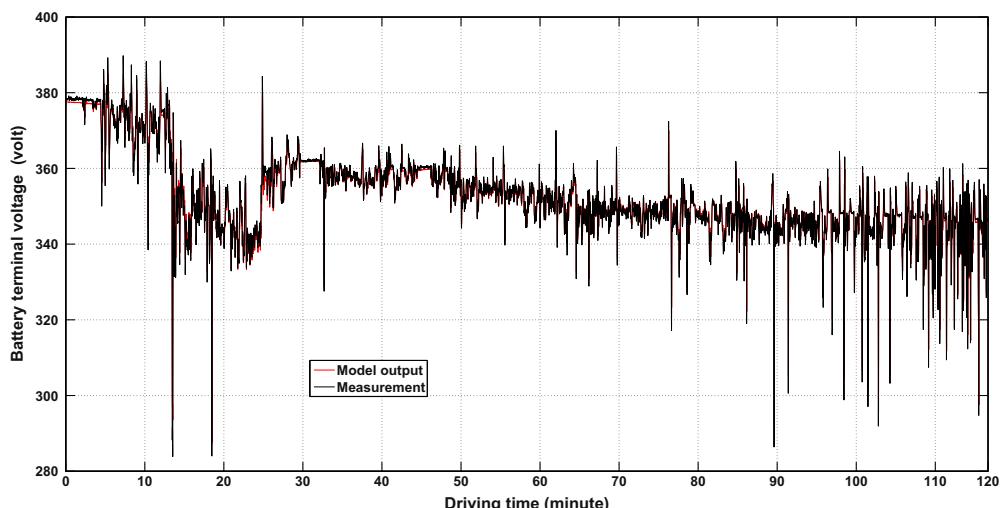


Fig. 12. Comparison of EV battery module model with measurements for driving case 1.

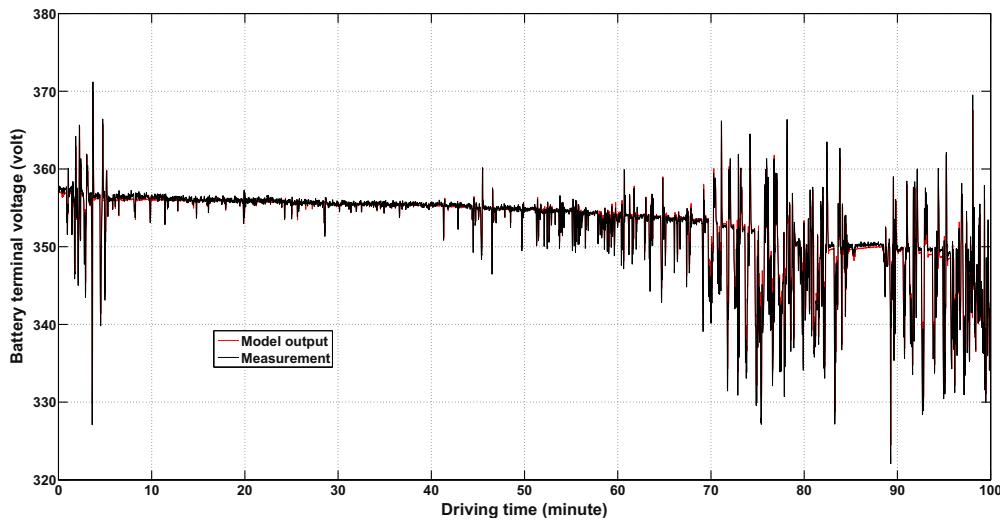


Fig. 13. Comparison of EV battery module model with measurements for driving case 2.

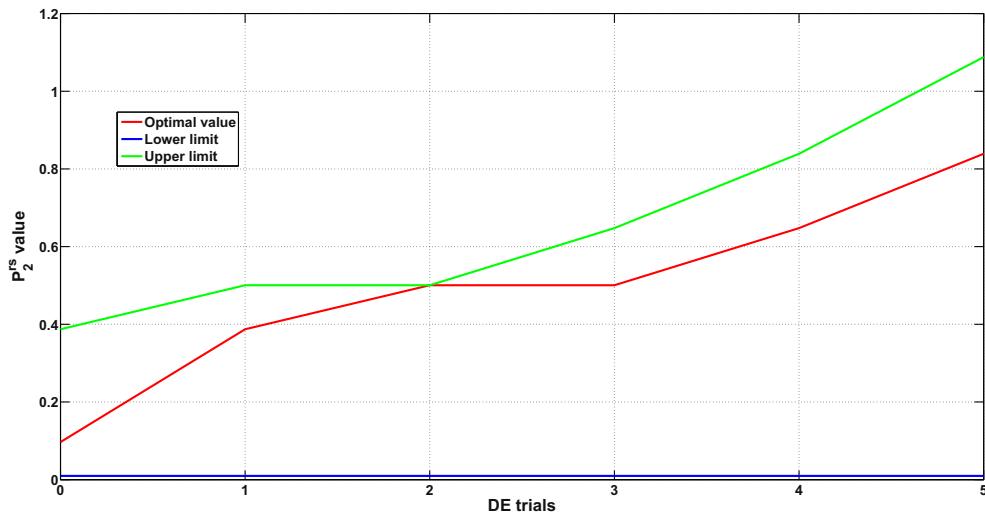


Fig. 14. An example of the evolution process of the parameter value and its boundaries.

example to illustrate the BES is shown in Fig. 14. Detailed parameter values are shown in Table 2.

It can be seen from Figs. 12 and 13 that there are good agreements between the simulation results and the measurements. Fig. 14 shows that the final parameter value is out of the original boundary, which proves the efficacy of the method. It is also found that the parameter values are quite different in the two cases. This

can be resulted from the factors such as ambient temperature and calendar life. This also shows the complexity of battery modelling.

6. Further discussion

From different case studies, it is seen that the proposed BES can efficiently update the variable boundaries. An argument here is to

Table 2
Parameterisation results from the EV battery.

Parameters	Initial SOC	Capacity	P_1^{rs}	P_2^{rs}	P_3^{rs}
Case 1	0.96	94.50	3.18×10^{-2}	0.48	0.16
Case 2	0.69	85.50	4.21×10^{-2}	0.84	0.15
Parameters	P_1^{r1}	P_2^{r1}	P_3^{r1}	P_1^{c1}	P_2^{c1}
Case 1	4.10×10^{-3}	3.79×10^{-6}	0.11	1.90×10^{-3}	0.35
Case 2	1.47×10^{-2}	3.11×10^{-6}	-1.63	1.60×10^{-3}	0.21
Parameters	P_1^{r2}	P_2^{r2}	P_3^{r2}	P_1^{c2}	P_2^{c2}
Case 1	1.80×10^{-6}	-1.30×10^{-2}	4.97	1.30×10^{-4}	0.21
Case 2	5.76×10^{-5}	-0.13	8.87×10^{-2}	-1.87×10^{-4}	5.20×10^{-3}

use wide variable boundaries together with a large population size. For example, regarding Schwefel function, it is viable to use 1 trial of DE with 1000 iterations and a population size of 50, and set $x_i \in [0.500]$. The required number of calculations of the objective function will be the same as the study presented in this paper. It is seen the final optimisation result is slightly better than the proposed approach [30]. However, it is worth mentioning that the approach is not intended to improve the optimality with appropriate variable boundaries. Instead, the BES is designed for those complex optimisation problems with uncertain initial boundaries, such as the problem studied in this work. Wide variable boundaries naturally appeal a large population size as well as a large number of iterations, which can be computationally expensive. The proposed approach provides an alternative where smaller population size can be employed while provides reasonably good results. Furthermore, it is also viable to extend the approach to enhance the search once reasonable variable boundaries are obtained.

7. Conclusion

The battery modelling can be essential to enable the applications towards future renewable energy systems and electrical vehicles. This paper proposes a boundary evolution strategy for battery model parameterisation approach based on differential evolution. The proposed BES addresses the uncertainties of parameter values in the battery model by updating the variable boundaries during the parameterisation to improve the final results. The proposed approach can estimate the internal impedances, initial SOC and the total battery capacity, and the efficacy is verified by various case studies.

Acknowledgements

The authors would like sincerely thank Mr. Anders Bro Pedersen for the kind help on the extraction of EV driving data.

References

- [1] P.M. Gomadam, J.W. Weidner, R.A. Dougal, R.E. White, *Journal of Power Sources* 110 (2002) 267–284.
- [2] S. Santhanagopalan, Q. Guo, P. Ramadass, R.E. White, *Journal of Power Sources* 156 (2006) 620–628.
- [3] A.P. Schmidt, M. Bitzer, A.W. Imre, L. Guzzella, *Journal of Power Sources* 195 (2010) 5071–5080.
- [4] N. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, A. Kojic, *Control Systems, IEEE* 30 (2010) 49–68.
- [5] L. Gao, S. Liu, R.A. Dougal, *IEEE Transactions on Components and Packaging Technologies* 25 (2002) 495–505.
- [6] M. Chen, Rincon-Mora, *IEEE Transactions on Energy Conversion* 21 (2006) 504–511.
- [7] C. Gould, C. Bingham, D. Stone, P. Bentley, *IEEE Transactions on Vehicular Technology* 58 (2009) 3905–3916.
- [8] R. Kroese, P. Krein, in: *Power Electronics Specialists Conference*, 2008, IEEE, pp. 1336–1342.
- [9] R.M. Spotnitz, *Electrochemical Society Interface* 14 (2005) 39–42.
- [10] Z. Salameh, M. Casacca, W. Lynch, *IEEE Transactions on Energy Conversion* 7 (1992) 93–98.
- [11] V. Johnson, *Journal of Power Sources* 110 (2002) 321–329.
- [12] M. Wu, C. Lin, Y. Wang, C. Wan, C. Yang, *Electrochimica Acta* 52 (2006) 1349–1357.
- [13] U.S. Kim, C.B. Shin, C. Kim, *Journal of Power Sources* 189 (2009) 841–846.
- [14] S. Al Hallaj, H. Maleki, J.S. Hong, J.R. Selman, *Journal of Power Sources* 83 (1999) 1–8.
- [15] P. Ramadass, B. Haran, R. White, B.N. Popov, *Journal of Power Sources* 123 (2003) 230–240.
- [16] B.Y. Liaw, R.G. Jungst, G. Nagasubramanian, H.L. Case, D.H. Doughty, *Journal of Power Sources* 140 (2005) 157–161.
- [17] O. Erdinc, B. Vural, M. Uzunoglu, in: *2009 International Conference on Clean Electrical Power*, 2009, pp. 383–386.
- [18] S. Abu-Sharkh, D. Doerffel, *Journal of Power Sources* 130 (2004) 266–274.
- [19] I. Bloom, B. Cole, J. Sohn, S. Jones, E. Polzin, V. Battaglia, G. Henriksen, C. Motloch, R. Richardson, T. Unkelhaeuser, D. Ingersoll, H. Case, *Journal of Power Sources* 101 (2001) 238–247.
- [20] M. Sitterly, L.Y. Wang, G. Yin, C. Wang, *IEEE Transactions on Sustainable Energy* 2 (2011) 300–308.
- [21] H. He, R. Xiong, H. Guo, *Applied Energy* 89 (2012) 413–420.
- [22] H. He, R. Xiong, J. Fan, *Energies* 4 (2011) 582–598.
- [23] J. Lee, J. Lee, O. Nam, J. Kim, B.H. Cho, H. Yun, S. Choi, K. Kim, J. Kim, S. Jun, in: *EPE-PEMC 2006: 12th International Power Electronics and Motion Control Conference, Proceedings*, 2007, pp. 1536–1540.
- [24] Y. Chiang, W. Sean, J. Ke, *Journal of Power Sources* 196 (2011) 3921–3932.
- [25] R. Storn, K. Price, *Differential Evolution—a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces* (1995). Technical Report TR-95-012, Berkeley, CA.
- [26] K.V. Price, in: *Biennial Conference of the North American Fuzzy Information Processing Society*, Berkeley, CA, USA, pp. 524–527.
- [27] S. Das, P. Suganthan, *IEEE Transactions on Evolutionary Computation* 15 (2011) 4–31.
- [28] M.M. Ali, P. Kaelo, *European Journal of Operational Research* 169 (Mar. 2006) 1176–1184.
- [29] D.E. Goldberg, J. Richardson, in: *Proceedings of the 2nd International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1987, pp. 41–49.
- [30] G.Y. Yang, Z.Y. Dong, K.P. Wong, *IEEE Transactions on Power Systems* 23 (2008) 514–522.
- [31] B. Vatchova, in: B. Reusch (Ed.), *Computational Intelligence, Theory and Applications, Advances in Soft Computing*, vol. 33, Springer, Berlin Heidelberg, 2005, pp. 17–24.
- [32] D. Whitley, K. Mathias, S. Rana, J. Dzubera, *Proceedings of the International Conference on Genetic Algorithms* (1995) 239–247.